



THE UNIVERSITY OF
CHICAGO

Section 10: Quadratic Programming

Reference: Chapter 16, Nocedal and
Wright.

10.1 GRADIENT PROJECTIONS FOR QPS WITH BOUND CONSTRAINTS

Projection

$$\min_x \quad q(x) = \frac{1}{2}x^T Gx + x^T c$$

- The problem: subject to $l \leq x \leq u$,
- Like in the trust-region case, we look for a Cauchy point, based on a projection on the feasible set.
- G does not have to be psd (essential for AugLag)
- The projection operator:

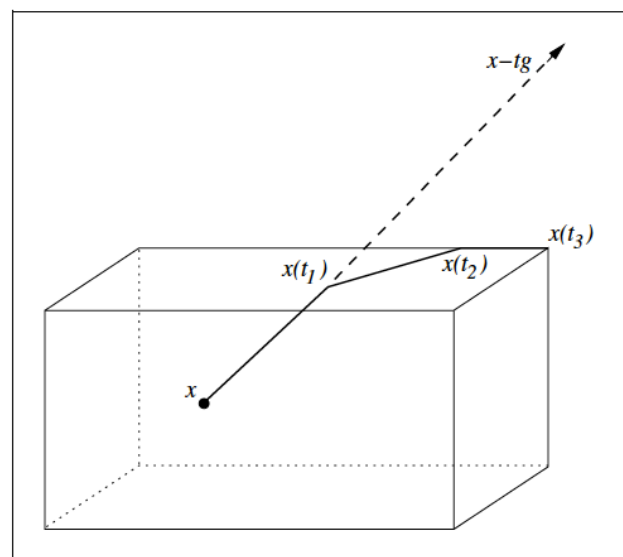
$$P(x, l, u)_i = \begin{cases} l_i & \text{if } x_i < l_i, \\ x_i & \text{if } x_i \in [l_i, u_i], \\ u_i & \text{if } x_i > u_i. \end{cases}$$

The search path

- Create a piecewise linear path which is feasible (as opposed to the linear one in the unconstrained case) by projection of gradient.

$$x(t) = P(x - tg, l, u),$$

$$g = Gx + c;$$



Computation of breakpoints

- Can be done on each component individually

$$\bar{t}_i = \begin{cases} (x_i - u_i)/g_i & \text{if } g_i < 0 \text{ and } u_i < +\infty, \\ (x_i - l_i)/g_i & \text{if } g_i > 0 \text{ and } l_i > -\infty, \\ \infty & \text{otherwise.} \end{cases}$$

- Then the search path becomes on each component:

$$x_i(t) = \begin{cases} x_i - t g_i & \text{if } t \leq \bar{t}_i, \\ x_i - \bar{t}_i g_i & \text{otherwise.} \end{cases}$$

Line Search along piecewise linear path

- Reorder the breakpoints eliminating duplicates and zero values to get

$$0 < t_1 < t_2 < \dots$$

- The path:

$$x(t) = x(t_{j-1}) + (\Delta t)p^{j-1}, \quad \Delta t = t - t_{j-1} \in [0, t_j - t_{j-1}],$$

- Whose direction is:

$$p_i^{j-1} = \begin{cases} -g_i & \text{if } t_{j-1} < \bar{t}_i, \\ 0 & \text{otherwise.} \end{cases}$$

Line Search (2)

- Along each piece, $[t_{j-1}, t_j]$ find the minimum of the quadratic

$$\frac{1}{2}x^T Gx + c^T x$$

- This reduces to analyzing a one dimensional quadratic form of t on an interval.
- If the minimum is on the right end of interval, we continue.
- If not, we found the local minimum and the Cauchy point.

Subspace Minimization

- Active set of Cauchy Point

$$\mathcal{A}(x^c) = \{i \mid x_i^c = l_i \text{ or } x_i^c = u_i\}.$$

- Solve subspace minimization problem

$$\begin{aligned} \min_x q(x) &= \frac{1}{2}x^T Gx + x^T c \\ \text{subject to } & x_i = x_i^c, \ i \in \mathcal{A}(x^c), \\ & l_i \leq x_i \leq u_i, \ i \notin \mathcal{A}(x^c). \end{aligned}$$

- No need to solve exactly. For example truncated CG with termination if one inactive variable reaches bound.

Gradient Projection for QP

Algorithm 16.5 (Gradient Projection Method for QP).

Compute a feasible starting point x^0 ;

for $k = 0, 1, 2, \dots$

 if x^k satisfies the KKT conditions for (16.68)

 stop with solution $x^* = x^k$;

 Set $x = x^k$ and find the Cauchy point x^c ;

 Find an approximate solution x^+ of (16.74) such that $q(x^+) \leq q(x^c)$
 and x^+ is feasible;

$x^{k+1} \leftarrow x^+$;

end (for)



Or, equivalently, if projection does not advance from 0.

Observations – Gradient Projection

- Note that the Projection – Active set solve loop must be iterated to optimality.
- What is the proper stopping criteria? How do we verify the KKT?
- Idea: When projection does not progress ! That is, on each component, either the gradient is 0, or the breakpoint is 0.

KKT conditions for Quadratic Programming with BC

10.2 QUADRATIC PROGRAMMING WITH EQUALITY CONSTRAINTS

Statement of Problem

- Problem:
$$\begin{aligned} \min_x \quad & q(x) \stackrel{\text{def}}{=} \frac{1}{2}x^T Gx + x^T c \\ \text{subject to} \quad & Ax = b, \end{aligned}$$

- Optimality Conditions:

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -p \\ \lambda^* \end{bmatrix} = \begin{bmatrix} g \\ h \end{bmatrix},$$

$$h = Ax - b, \quad g = c + Gx, \quad p = x^* - x.$$

- But, when is a solution of KKT a solution of the minimization problem?

10.2.1 DIRECT (FACTORIZATION APPROACHES)

Inertia of the KKT matrix

- Separate the eigenvalues of a symmetric matrix by sign. Define: $\text{inertia}(K) = (n_+, n_-, n_0)$.
- Result: (K=kkt matrix)

Let K be defined by (16.7), and suppose that A has rank m . Then

$$\text{inertia}(K) = \text{inertia}(Z^T G Z) + (m, m, 0).$$

Therefore, if $Z^T G Z$ is positive definite, $\text{inertia}(K) = (n, m, 0)$.

- But, how do I find inertia of K ?, ideally while finding the solution of the KKT system?

LDLT factorization

- Formulation: $P^T K P = L B L^T$,
- Solving the KKT system with it:

$$\begin{aligned} & \text{solve } Lz = P^T \begin{bmatrix} g \\ h \end{bmatrix} && \text{to obtain } z; \\ & \text{solve } B\hat{z} = z && \text{to obtain } \hat{z}; \\ & \text{solve } L^T \bar{z} = \hat{z} && \text{to obtain } \bar{z}; \\ & \text{set } \begin{bmatrix} -p \\ \lambda^* \end{bmatrix} = P\bar{z}. \end{aligned}$$

- Sylvester theorem: $\text{inertia}(A) = \text{inertia}(C^T D C)$
- And B should have $(n, m, 0)$ inertia !!!

10.2.2 EXPLICIT USE OF FEASIBLE SET

Reduction Strategies for Linear Constraints

- Idea: Use a special form of the Implicit Function Theorem

$$Y \in \mathbb{R}^{n \times m}; \quad Z \in \mathbb{R}^{n \times (n-m)} \quad [Y \mid Z] \in \mathbb{R}^{n \times n} \text{ is nonsingular, } AZ = 0.$$

- In turn, this implies that $A[Y \mid Z] = [AY \mid 0]$ and thus AY is full rank and invertible.
- We can thus parametrize the feasible set along the components in the range of Y and Z .

$$x = Yx_Y + Zx_Z,$$

Using the Y,Z parameterization

- This allows easy identification of the Y component of the feasible set:

$$x = Yx_Y + Zx_Z, \quad \longrightarrow \quad x_Y = (AY)^{-1}b. \quad \longrightarrow \quad x = Y(AZ)^{-1}b + Zx_Z$$

- The reduced optimization problem.

$$\min_{x_Z} f(Y(AZ)^{-1}b + Zx_Z).$$

How do we obtain a “good “ YZ parameterization?

- Idea: use a version of the QR factorization

$$Ax = b \Rightarrow A^T \Pi = \begin{bmatrix} \underbrace{Q_1}_{n \times m} & \underbrace{Q_2}_{n \times (n-m)} \end{bmatrix} \begin{bmatrix} \overset{m \times m}{\underbrace{\tilde{R}}} \\ 0 \end{bmatrix}$$

- After which, define

$$A^T \overset{m \times m}{\underbrace{\tilde{\Pi}}} = \begin{bmatrix} \underbrace{Q_1}_{n \times m} & \underbrace{Q_2}_{n \times (n-m)} \end{bmatrix} \begin{bmatrix} \overset{m \times m}{\underbrace{\tilde{R}}} \\ 0 \end{bmatrix} \Rightarrow Y = Q_1, Z = Q_2; Q_1^T Q_2 = 0_{m \times (n-m)} \Rightarrow$$

$$[AZ]^T \Pi = Q_2^T A^T \Pi = \begin{bmatrix} Q_1 & \underbrace{Q_2} \end{bmatrix} \begin{bmatrix} \overset{m \times m}{\underbrace{\tilde{R}}} \\ 0 \end{bmatrix} = 0_{(n-m) \times (m)}$$

$$[AY]^T \Pi = Q_1^T A^T \Pi = R \Rightarrow AY = [R \Pi^T]^T = \Pi R^T$$

10.2.3 NULL SPACE METHOD

Affine decomposition

- Ansatz ($AZ=0$):

$$p = Yp_Y + Zp_Z,$$

- Consequence:

$$\begin{array}{ll} \min_x & q(x) \stackrel{\text{def}}{=} \frac{1}{2}x^T Gx + x^T c \\ \text{subject to} & Ax = b, \end{array}$$

$$h = Ax - b, \quad g = c + Gx, \quad p = x^* - x.$$

$$(AY)p_Y = -h. \quad \longrightarrow \quad p_Y = -(AY)^{-1} h$$

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -p \\ \lambda^* \end{bmatrix} = \begin{bmatrix} g \\ h \end{bmatrix},$$

- We can work out the
Normal component as well:

$$-GYp_Y - GZp_Z + A^T \lambda^* = g$$

Normal component

- Multiply with Z transpose the equation:

$$(Z^T G Z) p_z = -Z^T G Y p_y - Z^T g.$$

- Use Cholesky, get p_z and then $p = Y p_y + Z p_z$,
(if not: second-order sufficient does not hold)
- Multiply first equation by Y^T to obtain

$$(AY)^T \lambda^* = Y^T (g + Gp),$$

- If I use QR, both are backsolves ($O(n^2)$)!!!

10.2.4 ITERATIVE METHODS: CG APPLIED TO REDUCED SYSTEM

Reduced problem

- Use affine decomposition:

$$x^* = Yx_Y + Zx_Z, \quad \longrightarrow \quad AYx_Y = b \Rightarrow x_Y = (AY)^{-1}b$$

- Reduced problem

$$\min_{x_Z} \frac{1}{2} x_Z^T Z^T G Z x_Z + x_Z^T c_Z, \quad c_Z = Z^T G Y x_Y + Z^T c.$$

- Associated linear system:

$$Z^T G Z x_Z = -c_Z.$$

Preconditioned Reduced CG

Algorithm 16.1 (Preconditioned CG for Reduced Systems).

Choose an initial point x_z ;

Compute $r_z = Z^T G Z x_z + c_z$, $g_z = W_{zz}^{-1} r_z$, and $d_z = -g_z$;

repeat

$$\alpha \leftarrow r_z^T g_z / d_z^T Z^T G Z d_z;$$

$$x_z \leftarrow x_z + \alpha d_z;$$

$$r_z^+ \leftarrow r_z + \alpha Z^T G Z d_z;$$

$$g_z^+ \leftarrow W_{zz}^{-1} r_z^+;$$

$$\beta \leftarrow (r_z^+)^T g_z^+ / r_z^T g_z;$$

$$d_z \leftarrow -g_z^+ + \beta d_z;$$

$$g_z \leftarrow g_z^+; \quad r_z \leftarrow r_z^+;$$

until a termination test is satisfied.

Preconditioner

- How do we do it?
- Idea:

$$W_{zz} \left(Z^T G Z \right) \approx I; \Rightarrow W_{zz} = Z^T H Z; W_{zz} \succ 0$$

10.2.5 ITERATIVE METHODS: PROJECTED CG

Rephrased reduced CG

- Use the projection matrix:

$$P = Z(Z^T H Z)^{-1} Z^T,$$

- The algorithm is equivalent to:

Algorithm 16.2 (Projected CG Method).

Choose an initial point x satisfying $Ax = b$;

Compute $r = Gx + c$, $g = Pr$, and $d = -g$;

repeat

$$\alpha \leftarrow r^T g / d^T G d;$$

$$x \leftarrow x + \alpha d;$$

$$r^+ \leftarrow r + \alpha G d;$$

$$g^+ \leftarrow P r^+;$$

$$\beta \leftarrow (r^+)^T g^+ / r^T g;$$

$$d \leftarrow -g^+ + \beta d;$$

$$g \leftarrow g^+; \quad r \leftarrow r^+;$$

until a convergence test is satisfied.

10.3 INTERIOR-POINT METHODS FOR CONVEX QUADRATIC PROGRAM

Setup

- The form of the problem solved:

$$\begin{array}{ll} \min_x & q(x) = \frac{1}{2}x^T Gx + x^T c \\ \text{subject to} & Ax \geq b, \end{array}$$

$$A = [a_i]_{i \in \mathcal{I}}, \quad b = [b_i]_{i \in \mathcal{I}}, \quad \mathcal{I} = \{1, 2, \dots, m\}.$$

Optimality Conditions

- In original form:

$$Gx - A^T \lambda + c = 0,$$

$$Ax - b \geq 0,$$

$$(Ax - b)_i \lambda_i = 0, \quad i = 1, 2, \dots, m,$$

$$\lambda \geq 0.$$

- With slacks:

$$Gx - A^T \lambda + c = 0,$$

$$Ax - y - b = 0,$$

$$y_i \lambda_i = 0, \quad i = 1, 2, \dots, m,$$

$$(y, \lambda) \geq 0.$$

Idea: define an “interior” path to the solution

- Define the perturbed KKT conditions as a nonlinear system:

$$F(x, y, \lambda; \sigma \mu) = \begin{bmatrix} Gx - A^T \lambda + c \\ Ax - y - b \\ \mathcal{Y} \Lambda e - \sigma \mu e \end{bmatrix} = 0,$$

- Solve successively while taking μ to 0 solves KKT:

$$F(x(\mu), y(\mu), \lambda(\mu); \mu, \sigma) = 0; y_i(\mu) \lambda_i(\mu) = \mu \sigma > 0$$

$$\mu \rightarrow 0 \Rightarrow (x(\mu), y(\mu), \lambda(\mu)) \rightarrow (x^*, y^*, \lambda^*) \text{ satisfies KKT}$$

How to solve it ?

- Solution: apply Newton's method for fixed μ :

$$\begin{bmatrix} G & 0 & -A^T \\ A & -I & 0 \\ 0 & \Lambda & \mathcal{Y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} -r_d \\ -r_p \\ -\Lambda \mathcal{Y} e + \sigma \mu e \end{bmatrix},$$

$$r_d = Gx - A^T \lambda + c, \quad r_p = Ax - y - b.$$

- New iterate:

$$(x^+, y^+, \lambda^+) = (x, y, \lambda) + \alpha(\Delta x, \Delta y, \Delta \lambda),$$

- Enforce: $(y^+, \lambda^+) > 0$:

How to PRACTICALLY solve it?

- Eliminate the slacks:

$$\begin{bmatrix} G & -A^T \\ A & \Lambda^{-1}\mathcal{Y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} -r_d \\ -r_p + (-y + \sigma \mu \Lambda^{-1}e) \end{bmatrix}.$$

- Eliminate the multipliers (note, the 22 block is diagonal and invertible)

$$(G + A^T \mathcal{Y}^{-1} \Lambda A) \Delta x = -r_d + A^T \mathcal{Y}^{-1} \Lambda [-r_p - y + \sigma \mu \Lambda^{-1}e],$$

- Solve (e.g by Cholesky if PD and modified Cholesky if not).